13th European Simulation Multiconference

ESM‘99

Warsaw, Poland, June 1-4, 1999

Using Java for Simulation

Michael Syrjakow

Institute for Computer Design
and Fault Tolerance (Prof. D. Schmid)
University of Karlsruhe, 76128 Karlsruhe
P.O. Box 6980, Germany
syrjakow@ira.uka.de

Outline

- Introduction
- Overview of Java
- Java-based simulation
  - Online simulation and animation
  - Simulation support packages
  - Distributed simulation
- Examples and experiences
- Conclusions

The Java Phenomenon

Java has emerged as a major force in the computing landscape

- as a programming language for the 90s
- as a new platform for the development of heterogeneous network-centric systems

in an astonishingly short period of time.

- Since May 1995 millions of copies of the JDK have been downloaded over the Internet
- Today almost every computer and telecommunication company has licensed Java technology from Sun Microsystems
- 14 000 people attended the 1998 JavaOne Conference

Implications and Questions for Computer Simulation

- Java has a number of features that have the potential to dramatically change the process used to **plan, build, execute, maintain, and distribute** simulation models
- **Important questions**
  - What are the benefits of Java and its related technologies for modelling and computer simulation?
  - Are there any disadvantages?
  - What is the state of the art of Java-based simulation?
  - What are the future prospects?
  - What shall we do with our existing simulation programs?

Overview of Java

Basic properties

- object-oriented
- familiar and simple
- safer and more reliable
- secure
- multithreaded
- interpreted and portable

$\Rightarrow$ **Programming language of the Web**

The Java Development Process

Java releases from Sun Microsystems

- January 1996

  JDK 1.0 for Solaris and Windows 95/NT

- May 1996

  JDK 1.0.2 for Solaris, Windows 95/NT, and Mac (3.66 MB)

- February 1997

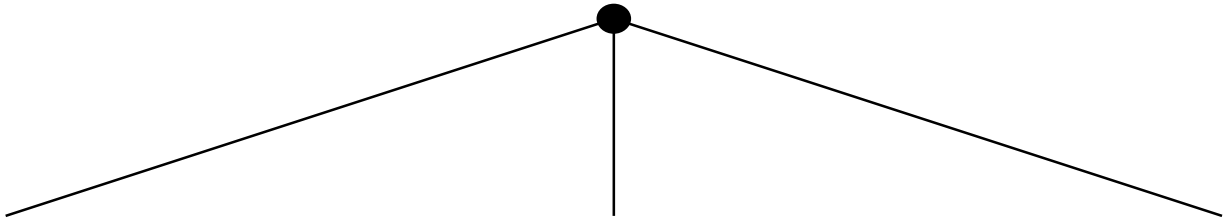
  JDK 1.1 (11.8 MB)

- December 1998

  JDK 1.2 (35.5 MB)

The Java Application Programming Interface (API)

Java libraries



Core libraries

Class libraries that every JDK must support

JDK 1.0: **8** core libraries
JDK 1.1: **22** core libraries
JDK 1.2: over **50** core libraries

Standard extensions to the JDK

Optional class libraries generally distributed with the JDK

APIs for specific hardware or application areas

Not necessarily intended to be distributed with the JDK

The Core Libraries

- Java language and utilities classes
- File input/output
- Graphical user interface
- Network communications
- Database connectivity support
- CORBA integration
- Security
- JavaBeans
- Java 2D
- ...

The Standard Extension Libraries

- Java 3D
- Java Media Framework
- Java Mail
- InfoBus
- Java Communications (serial and parallel port) API
- Java Help
- Enterprise JavaBeans
- Java Advanced Imaging
- Java Servlet API
- ...

Java APIs for Specific Hardware or Applications

- Personal Java
- Embedded Java
- Java Card
- Java Electronic Commerce Framework
- Jini
- These and other miscellaneous Java products and APIs

  http://java.sun.com/products/

Why is Java so Attractive?

- Object-orientation
- Powerful API
- Similarity to C/C++
- Platform-independence
- Close relationship to other key technologies
- Applet concept
- Available for free

Java-Based Simulation

Important areas

- Online simulation and animation
- Simulation support packages
- Distributed simulation

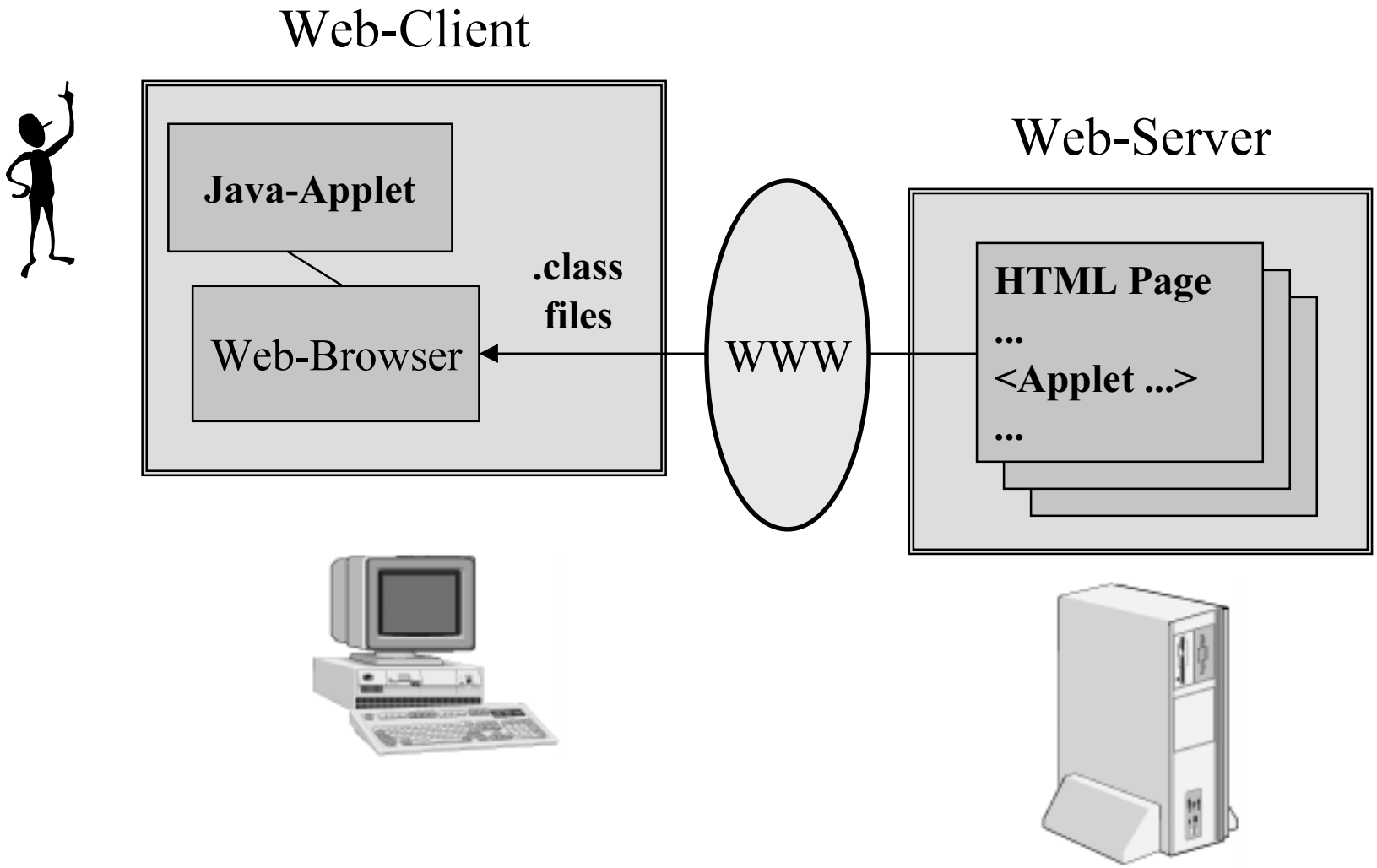
Additional information

http://ms.ie.org/websim/survey/survey.html

Online Simulation and Animation

- Simulations that are on the Web and available for execution
- Realisation alternatives
  - Client-side execution
    - Java applets
  - Server-side execution
    - CGI scripts
    - Java servlets

Client-Side Execution Using Java Applets



Java Applets

- **Advantages**
  - immediately executable anywhere in the World at any time on any platform without having to install anything

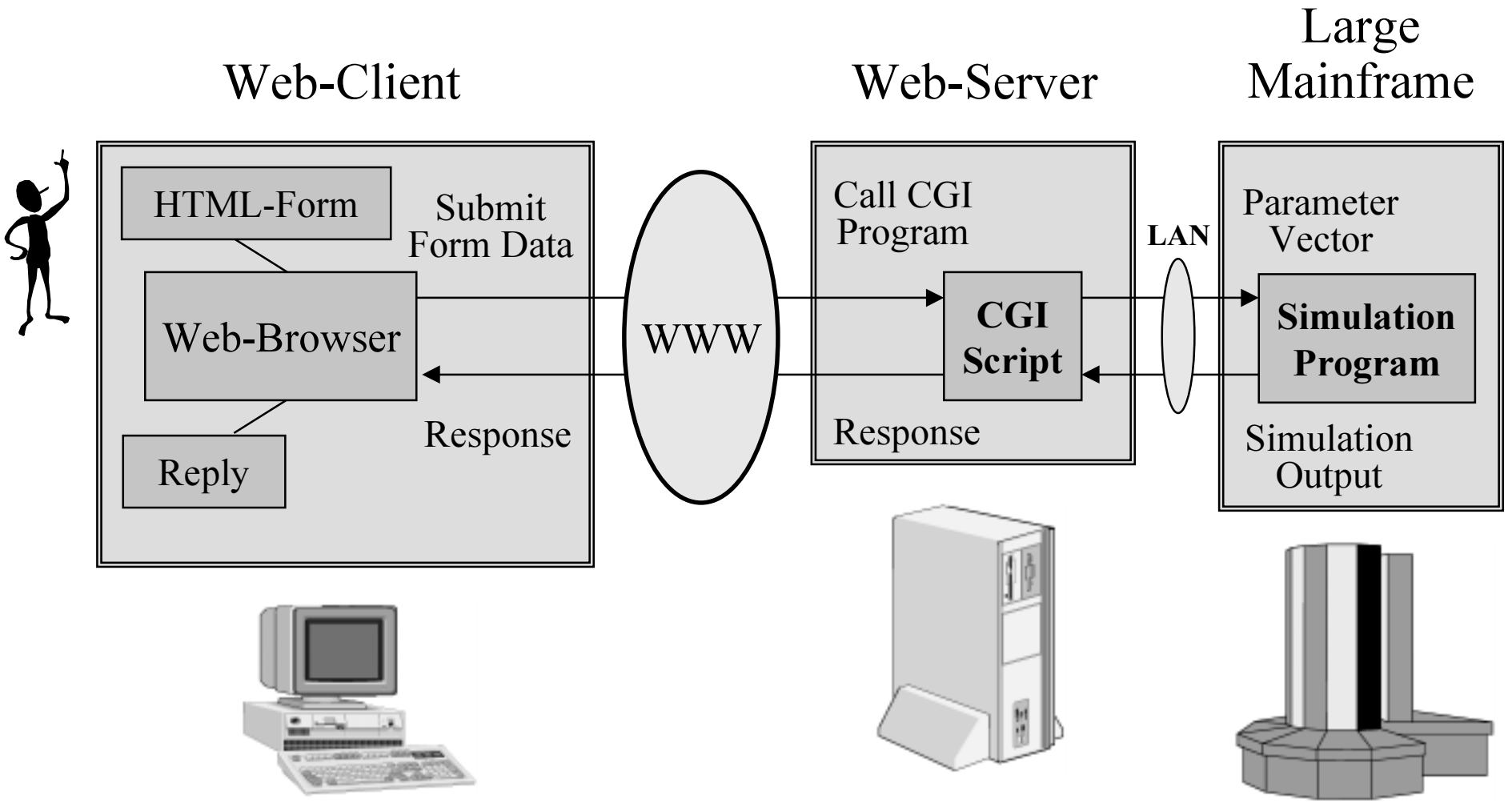
- **Disadvantages**
  - high network traffic
  - (re-)implementation in Java
  - extensive simulation runs require powerful clients
  - execution results may vary according to the applied browser
  - restricted access to the client system

Java Applets

- **Suitable for**
  - small- and medium-sized animations and simulations

- **Application fields**
  - teaching and education
  - demonstration of research results
  - sales promotion

Server-Side Execution Using CGI Scripts

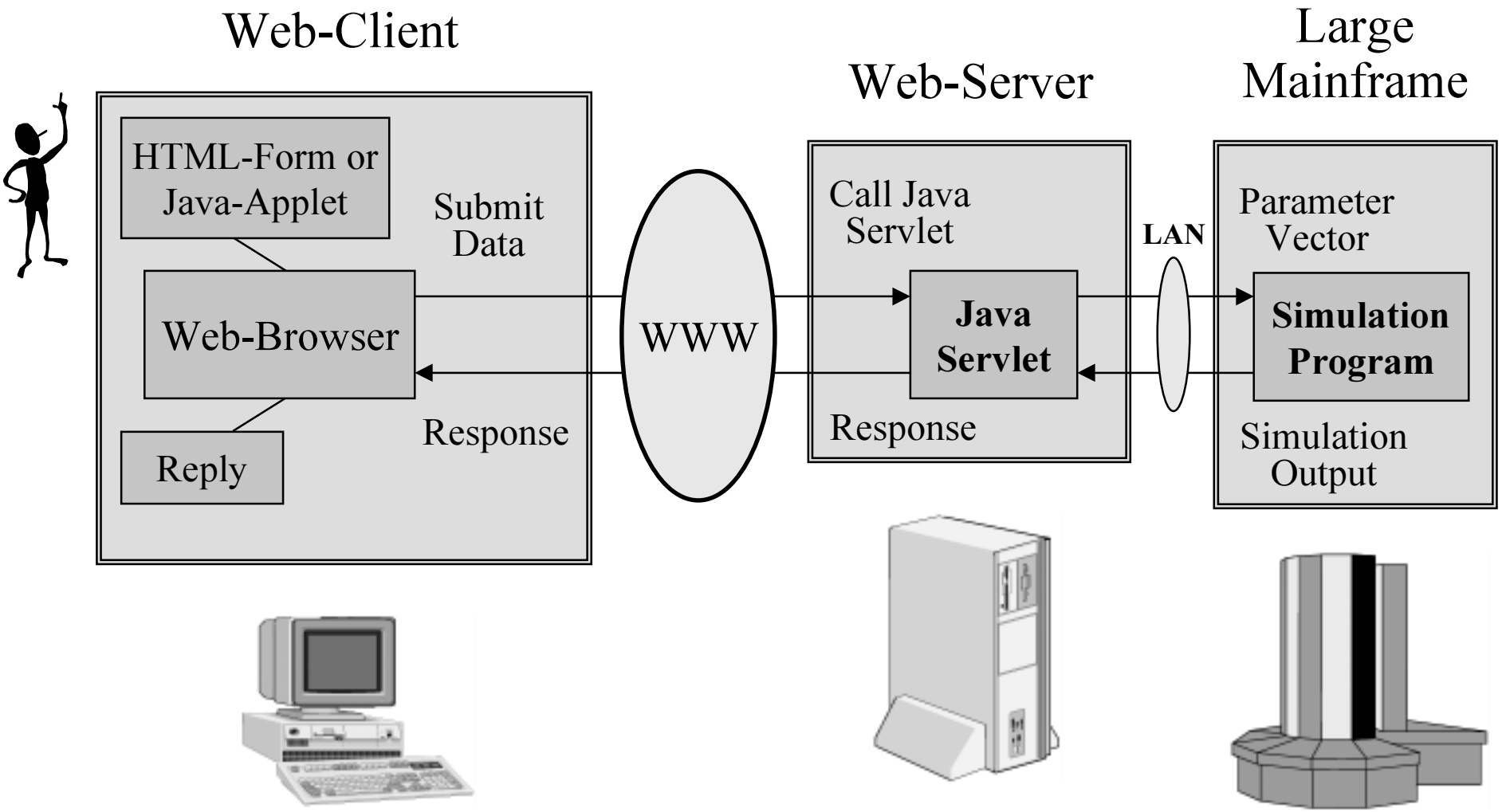


CGI-Scripts

- **Advantages**
  - thin clients
  - no reimplementation of existing simulation programs
  - small network traffic

- **Disadvantages**
  - no real-time animations („live“ simulations)
  - highly platform-dependent
  - difficult development, implementation, and fault detection
  - complex control of resource contention and consumption

Server-Side Execution Using Java Servlets



Java Servlets

- **Advantages**
  - thin clients
  - no reimplementation of existing simulation programs
  - small network traffic
  - platform-independence
  - much better performance and management characteristics than CGI scripts
  - simple implementation and usage
- **Disadvantages**
  - servlet capable Web-server

Simulation Support Packages

- Simjava
- Silk
- JavaSim
- JSIM
- DEVS-JAVA
- DESMO-J
- ...

Simjava

- Process based discrete event simulation package for Java
- Based on SIM++ (a discrete event simulation library for C++)
- Developed and distributed by

  The Institute for Computing Systems Architecture,

  Division of Informatics,

  University of Edinburgh.
- Freely available in full source code for download at

  http://www.dcs.ed.ac.uk/home/hase/simjava/

Simjava Packages

Simjava consists of three separate java packages

- simjava

  discrete event simulation library

- simanim

  package to provide animation

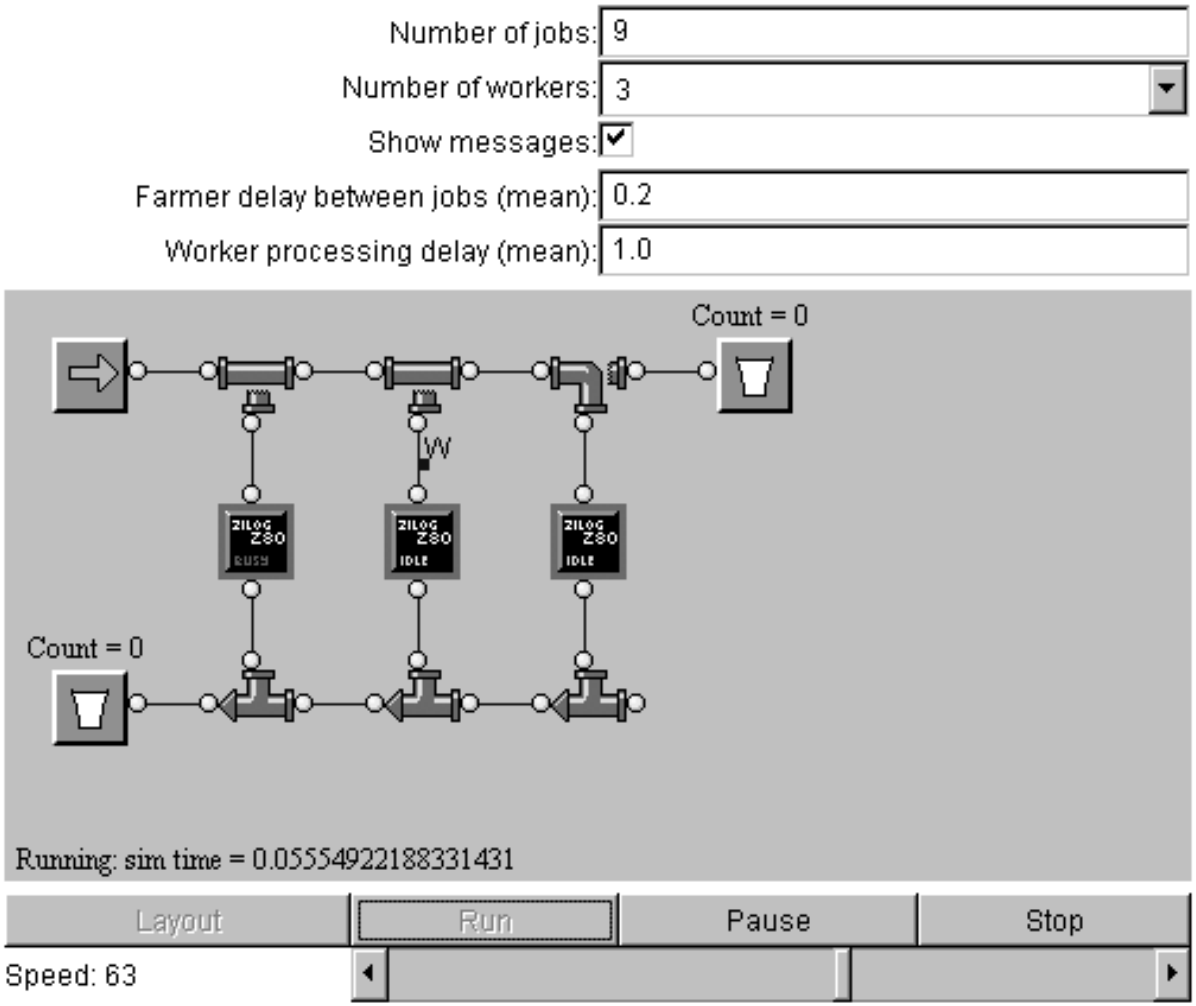
- simdiag

  collection of classes for displaying simulation results

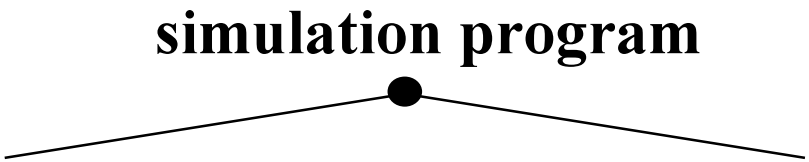
⇒ **basic elements necessary for building animated discrete event simulations in Java**

Simjava Example

A simple work farm



Performance Comparison



SIM++
C++ libraries

Simjava
Java libraries

Platform	Average execution time over 5 runs
C++ version under Solaris on Sun SPARCstation 5	**1538 ms**
Java application under Solaris on Sun SPARCstation 5	**12910 ms**
Java applet under Solaris/Netscape on Sun SPARCstation 5	**11214 ms**
Java applet under Windows NT/Netscape on Pentium 133 MHz	**9341 ms**

Distributed Simulation

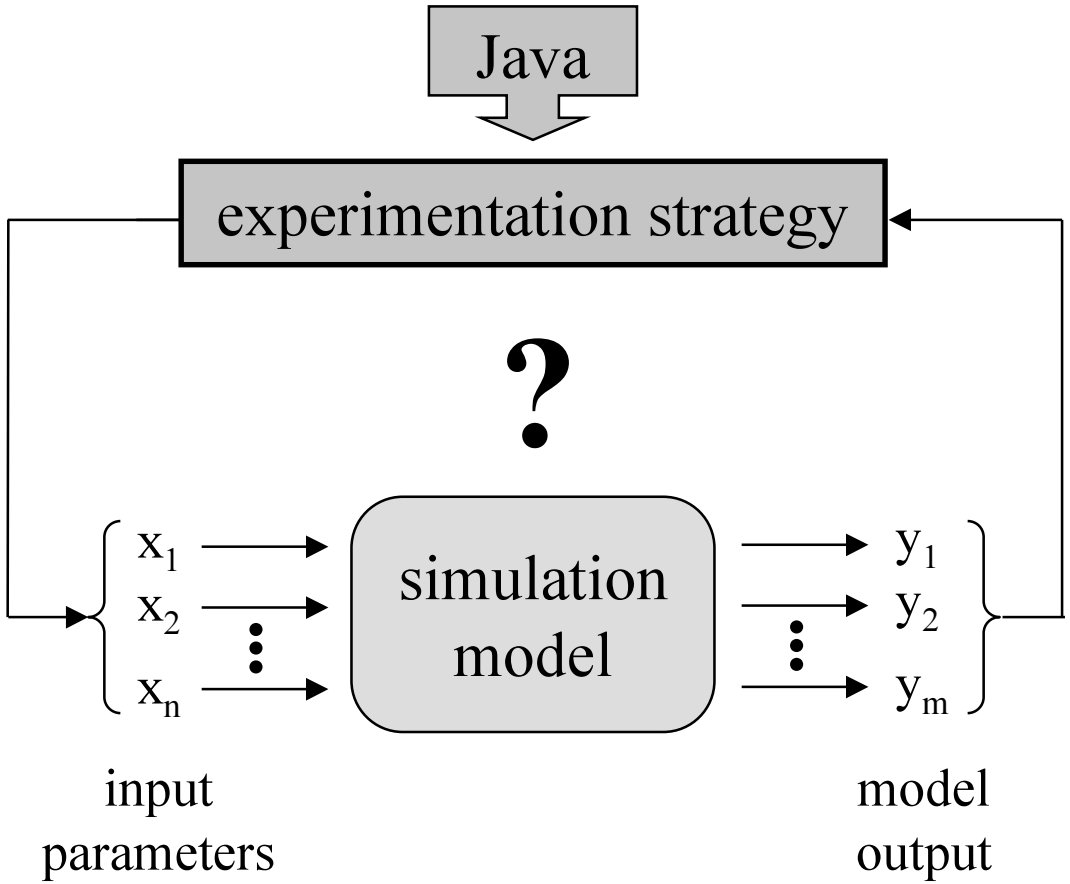
- **Basic communication requirements**
  - remote procedure call (RPC)
  - remote method invocation (RMI)
    - java.rmi
- **Infrastructures for object-oriented distributed computing**
  - component frameworks
    - ActiveX/(D)COM
    - Corba
    - Java Beans
  - agent platforms
    - Voyager
    - Aglets

Distributed Simulation Infrastructures and Architectures

- Aggregate Level Simulation Protocol (ALSP)
- Distributed Interactive Simulation (DIS) Protocol
- High Level Architecture (HLA)
  - Developed by the US Department of Defense under the leadership of the Defense Modeling and Simulation Office (DMSO)
  - Goal: Support of simulation reuse and interoperability
  - Detailed information at: http://hla.dmso.mil/

Examples and Experiences

Efficient strategies for experimentation with simulation models



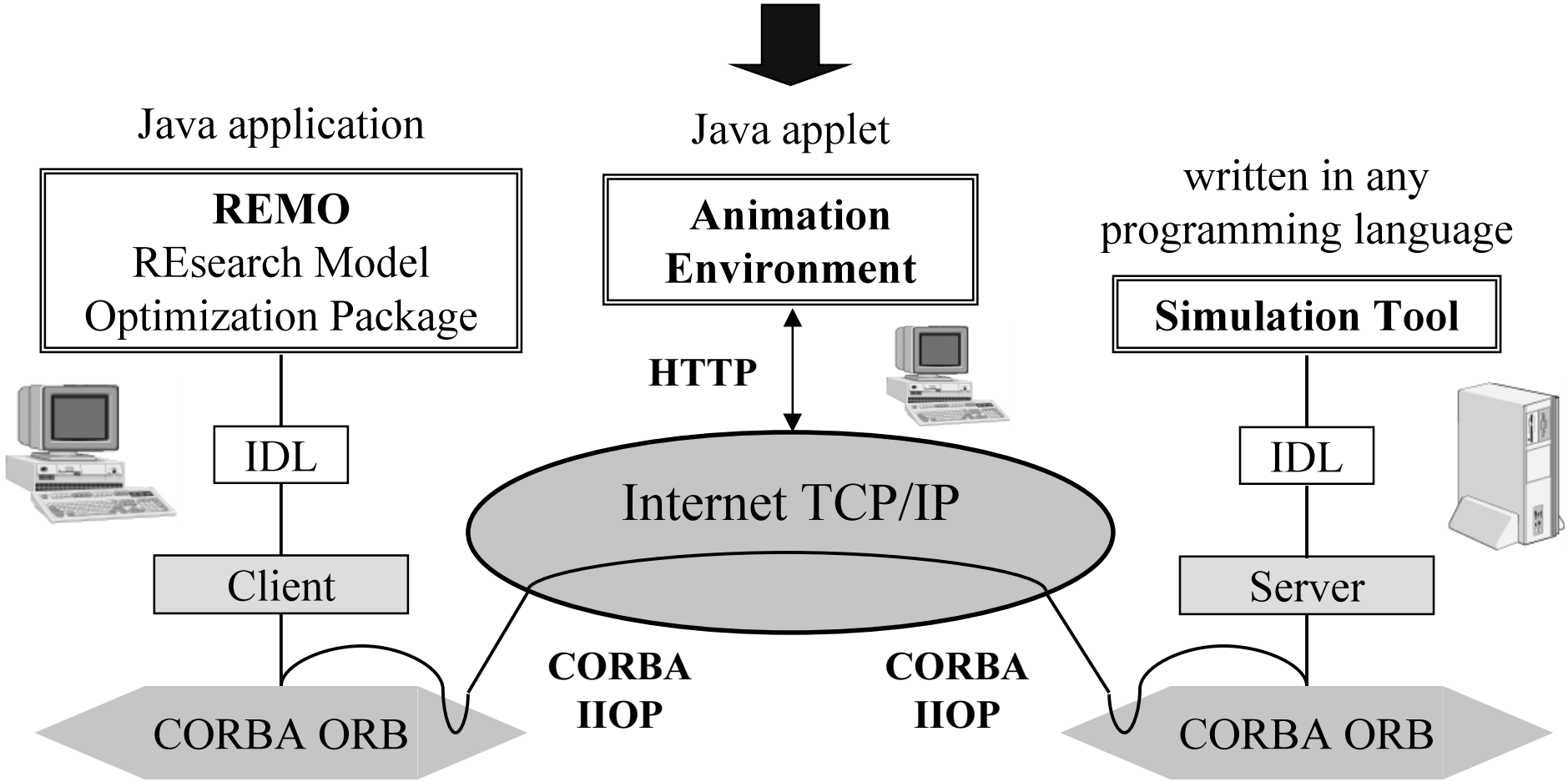
Possible Objectives

- Determination of sensitive/essential/critical parameters
- Search for optimal parameter settings
- Analysis of model behaviour
- Model validation

Advantages of Java

- object-oriented implementation
- platform-independent execution
- connection with existing simulation tools
- animation of experimentation trajectories

Scenario for Model Optimisation



Animation Environment for Direct Optimisation Methods

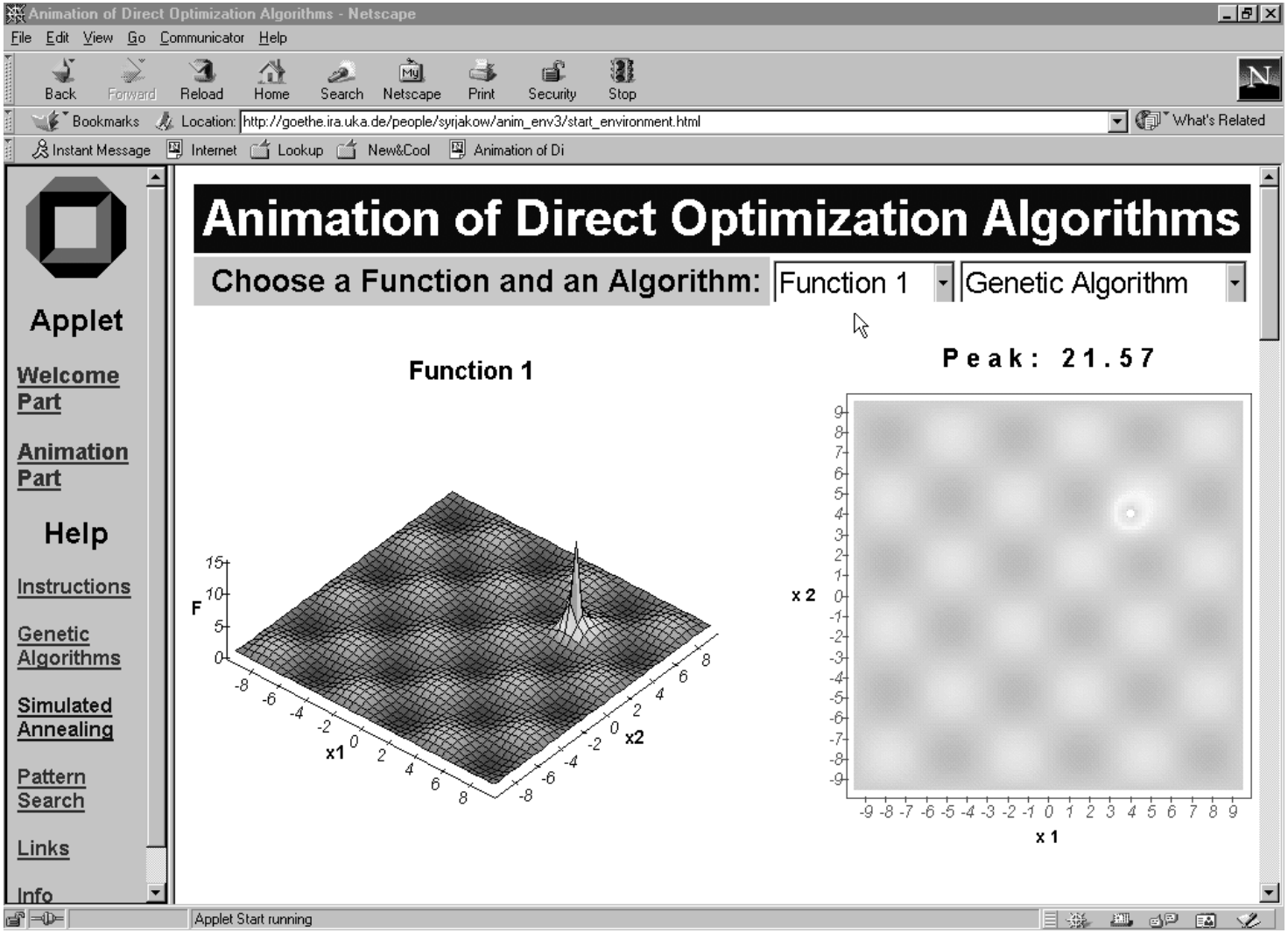
- **Well-known representatives of direct search methods**
  - Genetic Algorithms
  - Simulated Annealing
  - Pattern Search
- **Basic properties of direct search**
  - exclusive usage of goal function values to guide the optimisation process
  - additional analytical information is not required
- **Our goal:** Visualisation of the direct search process

Application Fields

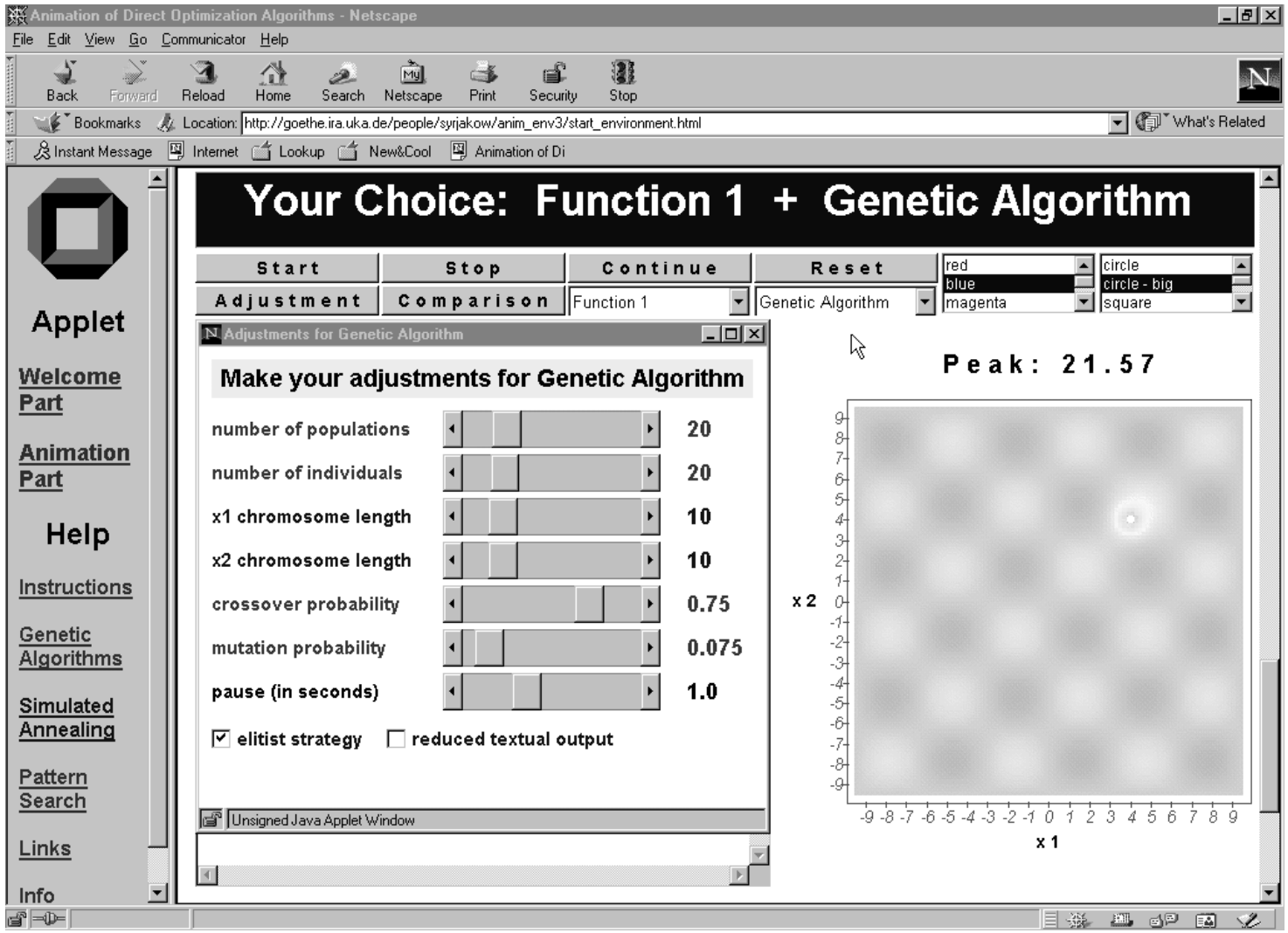
- **Research**
  - Support of the research work on stochastic search techniques
  - Illustrative demonstration of research results
- **Education**
  - Interactive teachware for
    - lectures or computer aided learning environments
    - inexperienced users/students to learn and understand how direct probabilistic optimisation works
- **Available at**

  http://goethe.ira.uka.de/people/syrjakow/animations.html

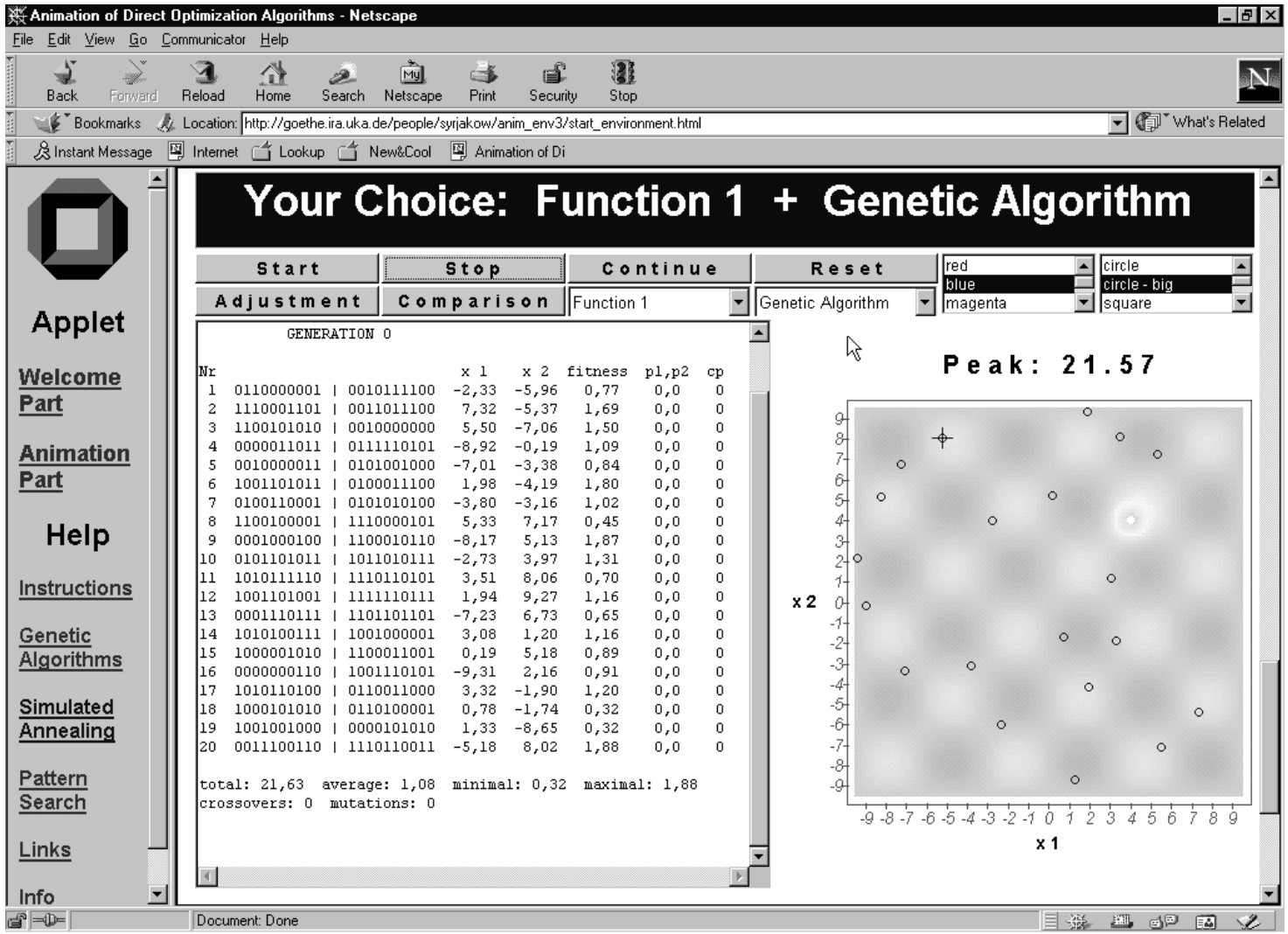
Welcome Part



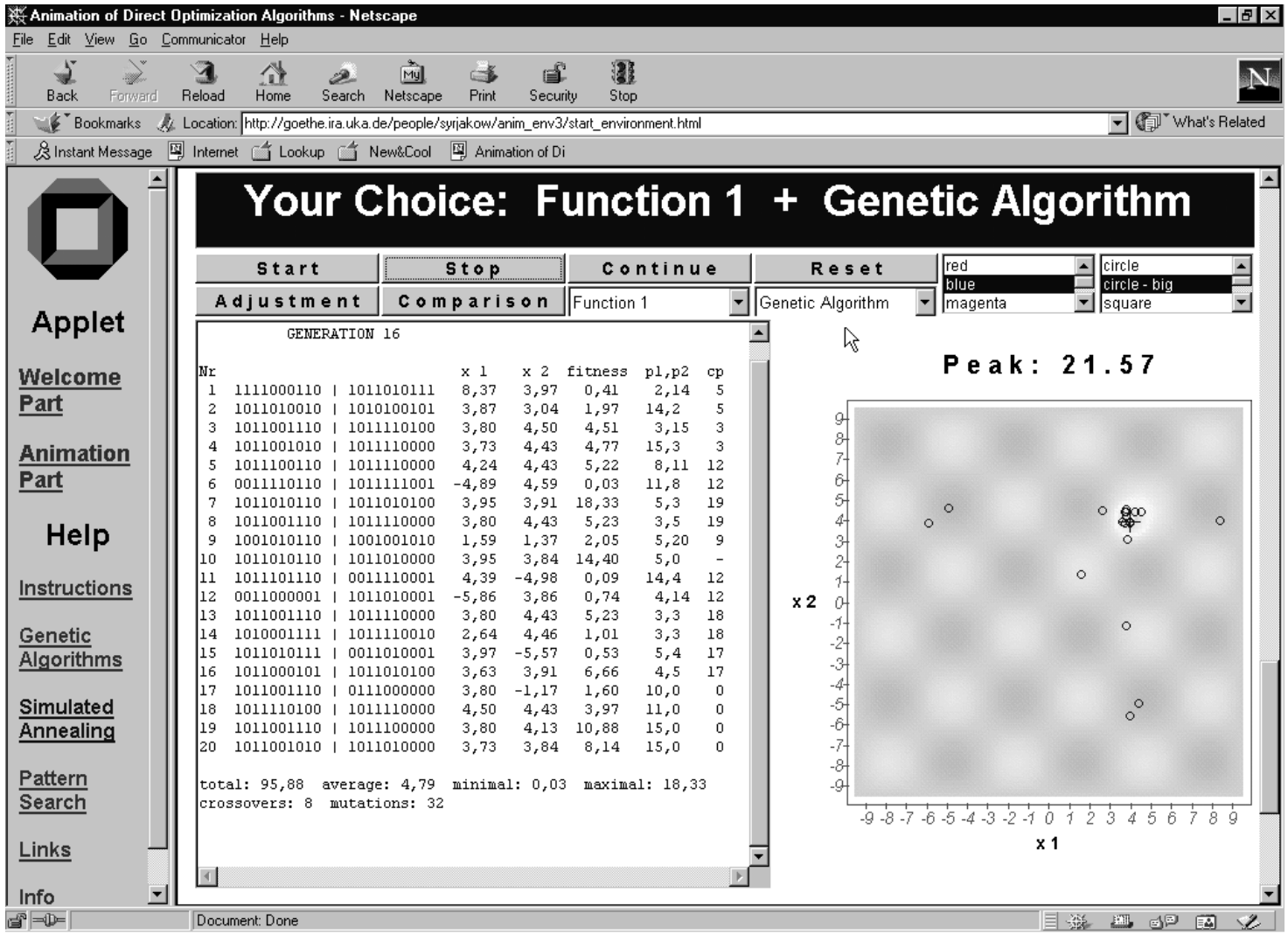
Animation Part



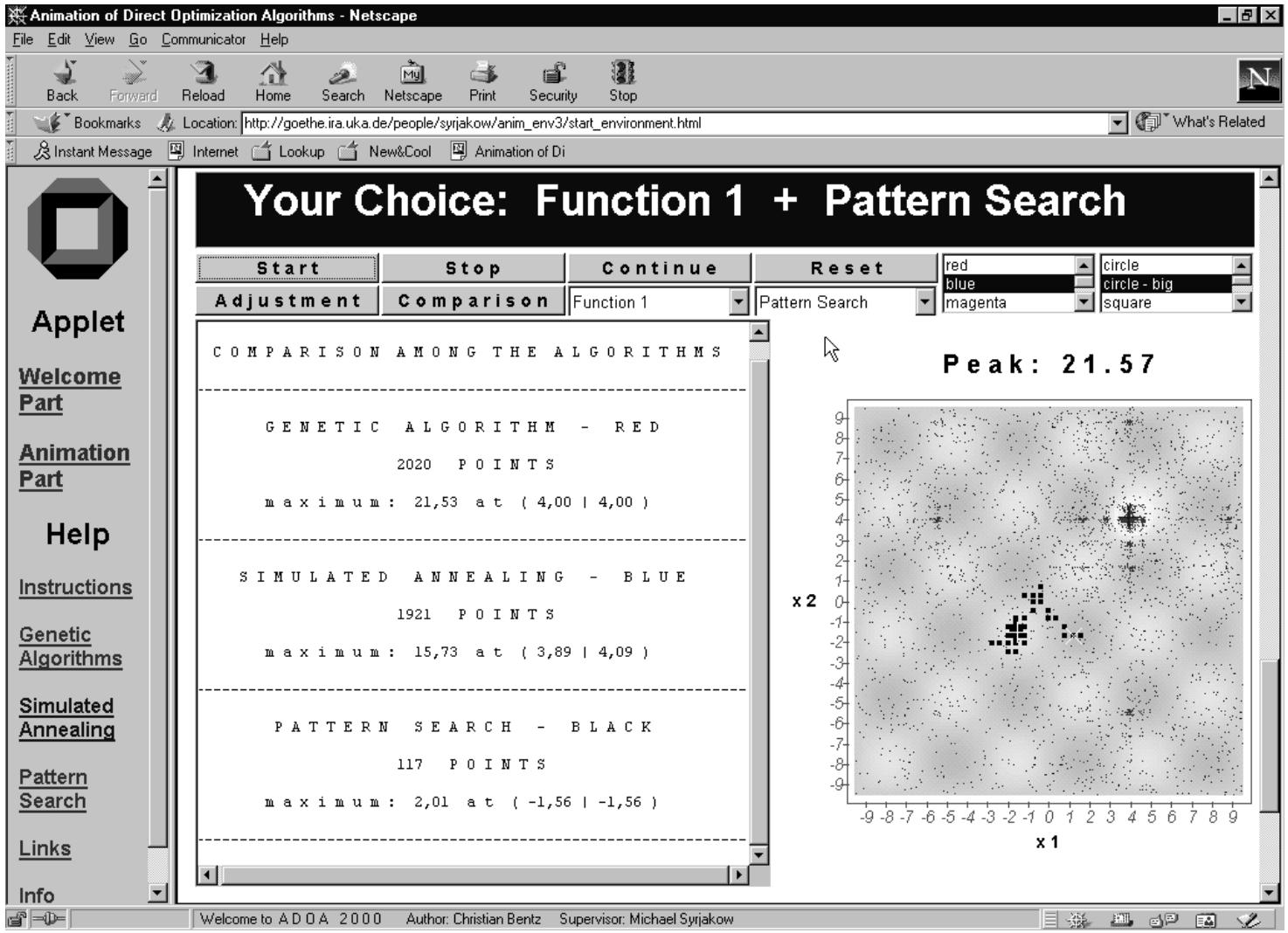
Animation Part



Animation Part



Animation Part



Conclusions

Benefits of Java

- object-orientation
- standardised libraries for all mainstream computing
- support for standardised components
- support for distributed processing
- support for multi-threaded programming
- platform independence
- Applet concept
- Close relationship to other key technologies

Conclusions

What has to be improved?

- execution speed
  - Just-In-Time Compiler
  - Java Processor
  - Java HotSpot Performance Engine
  - trend towards faster and cheaper hardware
- stability of the core API

Conclusions

Also of great importance will be

- Distributed objects/components
  - as subject and tool for distributed modelling and simulation
- Distributed simulation infrastructures and architectures
  - allowing reuse and interoperability across large numbers of different types of simulations
- Web-based simulation
  - as a rapidly emerging area of simulation research and development
- Application security
  - to guarantee a secure utilisation of Web-based simulations